Greedily Improving Our Own Centrality in A Network

Pierluigi Crescenzi, Gianlorenzo D'Angelo, Severini Lorenzo, Yllka Velaj

Greedily Improving Our Own Centrality in a Network

Pierluigi Crescenzi[1], Gianlorenzo D'Angelo[2(✉)], Lorenzo Severini[2], and Yllka Velaj[2]

[1] Department of Information Engineering, University of Florence, Viale Morgagni, 65, 50134 Florence, Italy
pierluigi.crescenzi@unifi.it

[2] Gran Sasso Science Institute (GSSI), Viale F. Crispi, 7, 67100 L'Aquila, Italy
{gianlorenzo.dangelo,lorenzo.severini,yllka.velaj}@gssi.infn.it

Abstract. The closeness and the betweenness centralities are two well-known measures of importance of a vertex within a given complex network. Having high closeness or betweenness centrality can have positive impact on the vertex itself: hence, in this paper we consider the problem of determining how much a vertex can increase its centrality by creating a limited amount of new edges incident to it. We first prove that this problem does not admit a polynomial-time approximation scheme (unless $P = NP$), and we then propose a simple greedy approximation algorithm (with an almost tight approximation ratio), whose performance is then tested on synthetic graphs and real-world networks.

1 Introduction

Looking for the most important vertices within a given complex network has always been one of the main goals in the field of real-world network analysis. Different measures of importance have been introduced in the literature, and several of them are related to the notion of "centrality" of a vertex. This latter notion, in turn, has been explicitly formalized in different ways: two of the most popular ways are closeness centrality and betweenness centrality (see, for example, [5]). The first one somehow measures the efficiency of a vertex while spreading information to all other vertices in its connected component, while the second one intuitively quantifies how much a vertex controls the information flow between all pairs of vertices in a graph. More formally, the closeness centrality of v is equal to the sum of the reciprocal of the distances to v from all other vertices, while the betweenness centrality of a given vertex v is the portion of the shortest paths between all pairs of vertices that pass through v.

Both closeness and betweenness centrality, however, are computationally expensive, since they require $O(nm)$ time [7] (in order to be computed for each vertex) which is clearly infeasible for networks with millions of vertices and edges (which is the "normal" size of many interesting real-world networks).

For this reason, several randomized and/or approximation algorithms have been proposed for the computation of these two centrality measures [9,20].

In this paper, instead, we consider a different problem related to the closeness and betweenness centrality, that is, the problem of identifying which "strategy" a vertex should adopt in order to increase its own centrality value. Indeed, increasing its own ranking in terms of centrality, can have positive consequences for the vertex. For example, in the field of author citation networks both closeness and betweenness centrality seem to be significantly correlated with citation counts (as it has been already observed in the case of collaboration networks) [23], while, in the field of transportation network analysis, the betweenness centrality seems to be positively related to the efficiency of an airport, as observed in [16] where a network of 57 European airports has been analyzed.

More specifically, we consider the problem of efficiently determining, for a given vertex v, the set of k edges entering v that, when added to the original directed graph, allows v to increase as much as possible its closeness (respectively, betweenness) centrality and its ranking according to this measure. We first prove that this problem is hard to be approximated within an approximation factor greater than $1-\frac{1}{3e}$ (respectively, $1-\frac{1}{2e}$), and we then show that a greedy approach yields an $(1-\frac{1}{e})$-approximation algorithm (for both closeness and betweenness). Successively, we present several experiments that we have performed (i) in order to evaluate how good is the approximation factor in the case of relatively small randomly generated graphs, and (ii) in order to apply the greedy approach to real-world citation and transportation networks. As a result of the first set of experiments, we have that the greedy algorithm seems to perform much better than the theoretical results, since it often computes an optimal solution and, in any case, it achieves an approximation factor significantly larger than $1-\frac{1}{e}$. By applying the greedy algorithm to real-world networks, instead, we observe that by adding very few edges a vertex can drastically increase its centrality measure and, hence, its ranking. For example, the first (respectively, second) author of this paper could pass from ranking 2540 to ranking 346 (respectively from ranking 6398 to 380), with just three citations. However, he has to convince Robert Tarjan, Christos Papadimitriou and Leslie Valiant (respectively, Richard Karp) to cite one of his papers. In the field of transportation networks, instead, the Paris Orly airport could increase its betweenness centrality (in the Easyjet connection network) by 218%, and pass from ranking 22 to ranking 15, with just three new connections from Ljubljana, Newquay, and Ponta Dalgada.

As far as we know, the problem analyzed in this paper has never been attacked before, even though similar problems have been studied for other centrality measures, i.e. page-rank [4,19], eccentricity [10], average distance [17], and some measures related to the number of paths passing through a given node [13]. Hence, we had no other algorithms to compare with. However, we also consider the naive approach of connecting the vertex with the top-k vertices in the centrality ranking and we experimentally show that the greedy algorithm significantly outperforms this simple heuristic, whenever $k > 1$.

1.1 Preliminary Definitions and Results

Let $G = (V, E)$ be a directed graph. For each node v, N_v denotes the set of in-neighbors of v, i.e. $N_v = \{u \mid (u, v) \in E\}$. Given two vertices s and t, we denote by d_{st}, σ_{st}, and σ_{stv} the distance from s to t in G, the number of shortest paths from s to t in G, and the number of shortest paths from s to t in G that contain v, respectively. Given a set S of edges not in E, we denote by $G(S)$ the graph augmented by adding the edges in S to G, i.e. $G(S) = (V, E \cup S)$. For a parameter x of G, we denote by $x(S)$ the same parameter in graph $G(S)$, e.g. the distance from s to t in $G(S)$ is denoted as $d_{st}(S)$. For each node v, the *closeness centrality* (also called *harmonic centrality* [5]) of v is defined as follows

$$c_v = \sum_{\substack{s \in V \setminus \{v\} \\ d_{sv} < \infty}} \frac{1}{d_{sv}},$$

while the *betweenness centrality* [5] of v is defined as

$$b_v = \sum_{\substack{s,t \in V \\ s \neq t; s,t \neq v \\ \sigma_{st} \neq 0}} \frac{\sigma_{stv}}{\sigma_{st}}.$$

The closeness and the betweenness centralities of a vertex clearly depend on the graph structure: if we augment a graph by adding a set of edges S, then the centrality of a vertex might change. Generally speaking, adding edges incident to some vertex v can only increase the centrality of v. We are interested in finding the set S of edges incident to a particular vertex v that maximizes such an increment. Therefore, we define the following optimization problem.

Maximum Closeness Improvement (MCI)	
Given:	A directed graph $G = (V, E)$; a vertex $v \in V$; and an integer $k \in \mathbb{N}$
Solution:	A set S of edges incident to v, $S = \{(u, v) \mid u \in V \setminus N(v)\}$, such that $\lvert S \rvert \leq k$
Goal:	Maximize $c_v(S)$

Analogously, we can define the **Maximum Betweenness Improvement** (in short, **MBI**), by referring to the betweeness centrality measure.

In this paper, we will use the the *maximum set coverage problem* [12] to derive approximation hardness results. Such problem is defined as follows.

Maximum Set Coverage (MSC)	
Given:	A set X; a family of subsets of X, $\mathcal{F} = \{S_1, S_2, \ldots S_{\lvert\mathcal{F}\rvert}\}$; and an integer k'
Solution:	A family $\mathcal{F}' \subseteq \mathcal{F}$ such that $\lvert\mathcal{F}'\rvert \leq k'$
Goal:	Maximize $s(\mathcal{F}') = \lvert \cup_{S_i \in \mathcal{F}'} S_i \rvert$

It has been shown [12] that MSC cannot be approximated within a factor greater than $1 - \frac{1}{e}$, unless $P = NP$. Moreover, the following greedy algorithm matches

Algorithm: GreedyImprovement
Input : A directed graph $G = (V, E)$; a vertex $v \in V$; and an integer $k \in \mathbb{N}$
Output: Set of edges $S \subseteq \{(u, v) \mid u \in V \setminus N_v\}$ such that $|S| \leq k$

```
1 S := ∅;
2 for i = 1, 2, ..., k do
3     foreach u ∈ V \ (N_v ∪ S) do Compute f_v(S ∪ {(u, v)});
4     u_max := arg max{f_v(S ∪ {(u, v)}) | u ∈ V \ (N_v ∪ S)};
5     S := S ∪ {(u_max, v)};
6 return S;
```

Fig. 1. The greedy centrality improvement algorithm (f_v denotes c_v or b_v)

such upper bound [18]: start with the empty set, and repeatedly add an element that gives the maximal marginal gain. The greedy algorithm can be extended to any *monotone submodular*[1] objective function defined on $\mathcal{F}$ thanks to the following result.

Theorem 1 ([18]). *For a non-negative, monotone submodular function f, let S be a set of size k obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the value of f. Then S provides a $\left(1 - \frac{1}{e}\right)$-approximation.*

In this paper, we exploit this result by showing that c_v and b_v are monotone and submodular w.r.t. the possible set of edges incident to v. Hence, the greedy algorithm reported in Fig. 1 (where f_v denotes either c_v or b_v) provides a $\left(1 - \frac{1}{e}\right)$-approximation. Note that the computational complexity of such algorithm is $O(k \cdot n \cdot g(n, m))$, where $g(n, m)$ is the complexity of computing either c_v or b_v.

2 Improving Closeness Centrality

We first prove that the problem of improving the closeness centrality of a vertex does not admit a polynomial-time approximation scheme.

Theorem 2. *Problem* MCI *cannot be approximated within a factor greater than $1 - \frac{1}{3e}$, unless $P = NP$.*

Proof. We give an L-reduction with parameters a and b [22]. In detail, we will give a polynomial-time algorithm that transforms any instance I_{MSC} of MSC into an instance I_{MCI} of MCI and a polynomial-time algorithm that transforms any solution S for I_{MCI} into a solution $\mathcal{F}'$ for I_{MSC} such that the following two

[1] For a ground set X, a function $f : 2^X \to \mathbb{N}$ is submodular if for any pair of sets $S \subseteq T \subseteq X$ and for any element $e \in X \setminus T$, $f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T)$ [18].

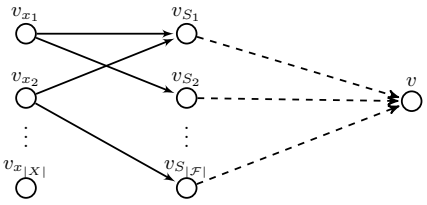

Fig. 2. The reduction used in Theorem 2 (in this example, $x_1 \in S_1$, $x_1 \in S_2$, $x_2 \in S_1$, and $x_2 \in S_{|\mathcal{F}|}$). The dashed edges denote those added in a solution.

conditions are satisfied for some values a and b:

$$OPT(I_{\text{MCI}}) \leq aOPT(I_{\text{MSC}}) \tag{1}$$

$$OPT(I_{\text{MSC}}) - s(\mathcal{F}') \leq b\,(OPT(I_{\text{MCI}}) - c_v(S))\,. \tag{2}$$

where OPT denotes the optimal value of an instance of an optimization problem. If the above conditions are satisfied and there exists a α-approximation algorithm for MCI, then there exists a $(1-ab(1-\alpha))$-approximation algorithm for MSC [22]. Since MSC is hard to approximate within a factor greater than $1-\frac{1}{e}$, then $1-ab(1-\alpha) < 1-\frac{1}{e}$, unless $P = NP$. This implies that $\alpha < 1-\frac{1}{abe}$.

Given an instance $I_{\text{MSC}} = (X, \mathcal{F}, k')$ of MSC, we define an instance $I_{\text{MCI}} = (G, v, k)$ of MCI as follows (see Fig. 2): $k = k'$ and $G = (V, E)$, where $V = \{v\} \cup \{v_{x_i} \mid x_i \in X\} \cup \{v_{S_j} \mid S_j \in \mathcal{F}\}$ and $E = \{(v_{x_i}, v_{S_j}) \mid x_i \in S_j\}$.

Without loss of generality, we can assume that any solution S of MCI contains only edges (v_{S_j}, v) for some $S_j \in \mathcal{F}$. In fact, if a solution does not satisfy this property, then we can improve it in polynomial time by repeatedly applying the following rule: if S contains an edge (v_{x_i}, v), for some $x_i \in X$, then exchange such edge with an edge (v_{S_j}, v) such that $(v_{S_j}, v) \notin S$ (note that such an edge must exist, since otherwise $|\mathcal{F}| \leq k = k'$ and I_{MSC} could be easily solved). The above rule does not decrease the value of $c_v(S)$: indeed, if we exchange an edge (v_{x_i}, v) with an edge (v_{S_j}, v) such that $(v_{S_j}, v) \notin S$, then the closeness centrality of v decreases by either 1 or $\frac{1}{2}$ (because of the deletion of (v_{x_i}, v)) but certainly increases by 1 (because of the insertion of (v_{S_j}, v)).

Given a solution S of MCI, let $\mathcal{F}'$ be the solution of MSC such that $S_j \in \mathcal{F}'$ if and only if $(v_{S_j}, v) \in S$. We now show that $c_v(S) = \frac{1}{2}s(\mathcal{F}') + k$. To this aim, let us note that the distance from a vertex v_{x_i} to v is equal to 2 if an edge (x_{S_j}, v) such that $x_i \in S_j$ belongs to S, and it is ∞ otherwise. Similarly, the distance from a vertex v_{S_j} to v is equal to 1 if $(x_{S_j}, v) \in S$, and it is ∞ otherwise. Moreover, the set of elements x_i of X such that $d_{v_{x_i}v}(S) < \infty$ is equal to $\{x_i \mid x_i \in S_j \wedge (v_{S_j}, v) \in S\} = \bigcup_{S_j \in \mathcal{F}'} S_j$. Therefore,

$$c_v(S) = \sum_{\substack{s \in V \setminus \{v\} \\ d_{sv}(S) < \infty}} \frac{1}{d_{sv}(S)} = \sum_{\substack{x_i \in X \\ d_{v_{x_i} v}(S) < \infty}} \frac{1}{d_{v_{x_i} v}(S)} + \sum_{\substack{S_j \in \mathcal{F} \\ d_{v_{S_j} v}(S) < \infty}} \frac{1}{d_{v_{S_j} v}(S)}$$

$$= \frac{1}{2} |\{x_i \in X \mid d_{v_{x_i} v}(S) < \infty\}| + |\{S_j \in \mathcal{F} \mid d_{v_{S_j} v}(S) < \infty\}|$$

$$= \frac{1}{2} \left| \bigcup_{S_j \in \mathcal{F}'} S_j \right| + |\{S_j \mid (v_{S_j}, v) \in S\}| = \frac{1}{2} s(\mathcal{F}') + k' = \frac{1}{2} s(\mathcal{F}') + k.$$

It follows that Conditions (1) and (2) are satisfied for $a = \frac{3}{2}$ and $b = 2$. Indeed, $OPT(I_{\text{MCI}}) = \frac{1}{2} OPT(I_{\text{MSC}}) + k \leq \frac{3}{2} OPT(I_{\text{MSC}})$, where the inequality is due to the fact that $OPT(I_{\text{MSC}}) \geq k$, since otherwise the greedy algorithm would find an optimal solution for I_{MSC}. Moreover, $OPT(I_{\text{MSC}}) - s(\mathcal{F}') = 2\,(OPT(I_{\text{MCI}}) - k) - 2\,(c_v(S) - k) = 2\,(OPT(I_{\text{MCI}}) - c_v(S))$. The theorem follows by plugging the values of a and b into $\alpha < 1 - \frac{1}{abe}$. □

2.1 Greedy Algorithm and Submodularity

We now prove that the GreedyImprovement algorithm provides a $(1 - \frac{1}{e})$-approximation for the MCI problem. To this aim, because of Theorem 1, it suffices to prove that the closeness centrality measure is monotone and submodular.

Theorem 3. *For each vertex v, function c_v is monotone and submodular with respect to any feasible solution for* MCI.

Proof. To show that c_v is monotone increasing, it is enough to observe that for each solution S to MCI, each vertex u such that $(u, v) \notin E \cup S$, and each $s \in V \setminus \{v\}$ such that $d_{sv}(S \cup \{(u, v)\}) \neq \infty$, then $d_{sv}(S \cup \{(u, v)\}) \leq d_{sv}(S)$ and therefore $\frac{1}{d_{sv}(S \cup \{(u,v)\})} \geq \frac{1}{d_{sv}(S)}$. We now show that for each pair S and T of solutions to MCI such that $S \subseteq T$ and for each vertex u such that $(u, v) \notin T \cup E$,

$$c_v(S \cup \{(u, v)\}) - c_v(S) \geq c_v(T \cup \{(u, v)\}) - c_v(T).$$

To simplify notation, we assume that $\frac{1}{d_{st}(X)} = 0$ whenever $d_{st}(X) = \infty$, for any solution X to MCI. We prove that each term of c_v is submodular, that is, that, for each vertex $s \in V \setminus \{v\}$ such that $d_{sv}(T \cup \{(u, v)\}) \neq \infty$, we show that

$$\frac{1}{d_{sv}(S \cup \{(u, v)\})} - \frac{1}{d_{sv}(S)} \geq \frac{1}{d_{sv}(T \cup \{(u, v)\})} - \frac{1}{d_{sv}(T)}. \tag{3}$$

Let us consider the shortest paths from s to v in $G(T \cup \{(u, v)\})$. The following two cases can arise:

1. The last edge of a shortest path from s to v in $G(T \cup \{(u, v)\})$ is (u, v) or belongs to $S \cup E$. In this case, such a path is a shortest path also in $G(S \cup \{(u, v)\})$, as it cannot contain edges in $T \setminus S$. Then, $d_{sv}(S \cup \{(u, v)\}) = d_{sv}(T \cup \{(u, v)\})$ and $\frac{1}{d_{sv}(S \cup \{(u,v)\})} = \frac{1}{d_{sv}(T \cup \{(u,v)\})}$. Moreover, $d_{sv}(S) \geq d_{sv}(T)$ and, therefore, $-\frac{1}{d_{sv}(S)} \geq -\frac{1}{d_{sv}(T)}$.

2. The last edge of all shortest paths from s to v in $G(T \cup \{(u,v)\})$ belongs to $T \setminus S$. In this case, $d_{sv}(T) = d_{sv}(T \cup \{(u,v)\})$ and, therefore, $\frac{1}{d_{sv}(T \cup \{(u,v)\})} - \frac{1}{d_{sv}(T)} = 0$. As $\frac{1}{d_{sv}(S)}$ is monotone increasing, then $\frac{1}{d_{sv}(S \cup \{(u,v)\})} - \frac{1}{d_{sv}(S)} \geq 0$.

In both cases, we have that the inequality (3) is satisfied and, hence, the theorem follows. □

2.2 Experimental Evaluation

We conducted two types of experiments: in the first type we evaluate the quality of the solution produced by the greedy algorithm by measuring the approximation ratio on several randomly generated networks; in the second type we measured the improvement in the value of closeness of v and in the closeness ranking of v within the network (these latter experiments are conducted on three real-world networks). All our experiments have been performed on a computer equipped with an AMD Opteron 6376 CPU with 16 cores clocked at 2.30GHz and 64GB of main memory, and our programs have been implemented in `C++` (gcc compiler v4.8.2 with optimization level O3).

We measured the approximation ratio of the greedy algorithm on four types of randomly generated networks, namely directed Preferential Attachment (in short, `PA`) [6], Erdős-Rényi (in short, `ER`) [11], Copying (in short, `COPY`) [14], and Compressible Web (in short, `COMP`) [8]. The size of the graphs is reported in Table 1. For each combination (n, m), we generated five random graphs and used five vertices as v. These vertices have been chosen on the basis of their original closeness ranking: in particular, we divided the list of vertices sorted by their original ranking in five parts and choose the vertices in the boundaries. We denote by $v_{X\%}$ the vertex on the boundary of the top Xth percentile (e.g. $v_{25\%}$ is a vertices on the boundary of the top 25th percentile).

In the experiments, we measured the ratio between the value of the solution found by the greedy algorithm and the optimal value computed by using an Integer Program (in short, IP). We solved IP by using the GLPK solver [3]. However, since solving IP requires long time on large instances, in some cases we used the solution to the Linear Relaxation (in short, LP) of IP as an upper bound to the optimal value. In these cases, the ratio is obtained by using the LP upper bound as a denominator, and therefore it represents a lower bound to the actual approximation ratio.

The results are reported in Table 1, where we show the number of times that the approximation ratio is equal to one and the minimum ratio obtained. The experiments clearly show that the measured approximation ratio is by far better than the theoretical one proven in the previous section. In fact, in more than 91% cases, the greedy algorithm found an optimal solution, and in the worst case the ratio is 0.9694.

For the second type of experiments we used real-world citation networks obtained by the Arnetminer database [1]. In such networks, there is a vertex for each author and an edge from vertex x to vertex y if the author corresponding to vertex x cited in his paper one paper written by the author corresponding to y.

Table 1. Closeness centrality: comparison between the greedy algorithm and the optimum (or an upper bound to the optimum). The first three columns report the type and size of the graphs; the fourth column reports the relative number of times that the greedy algorithm finds an optimal solution. The fifth column reports the minimum measured approximation ratio. The last column indicates whether the optimum has been found by using the integer program (IP) or its linear relaxation (LP), in the latter case it is an upper bound to the optimum.

Network	$n = \|V\|$	$m = \|E\|$	OPT%	**Min Approx. Ratio**	**IP/LP**
PA	100,500,1000	$\approx 1.3 \times n$	93.25	0.9831	IP
ER	100	200, 500, 1000	88.60	0.9788	IP
ER	500	5000, 12500, 25000	74.28	0.9694	LP
COMP	100	200, 500, 1000	99.88	0.9764	LP
COMP	500	5000, 12500, 25000	99.47	0.9854	LP
COPY	100	200, 500, 1000	97.48	0.9885	LP
COPY	500	5000, 12500, 25000	89.65	0.9697	LP

We parsed the Arnetminer database in order to select three sub-networks induced by the authors that published at least a paper in one of the main conferences or a journals in (i) algorithms (THEORY network, with 9274 vertices and 130419 edges), social network analysis (SN network, with 3666 vertices and 32413 edges), and computer science education (CSE network, with 3680 vertices and 35691 edges). As in the previous experiment, for each graph, we used five vertices as v. The value of k ranges from 1 to 100.

The results for THEORY are plotted in Fig. 3 (the results for the other networks are similar). In the two top charts we plot the closeness centrality and the ranking of vertex v as a function of k. We observe that any vertex become central by adding just few edges. For example a vertex with the smallest closeness centrality which initially has closeness 0 and is ranked 6398, improves its closeness and ranking to 2705.97 and 215, respectively, by adding only 10 edges, and it is among the top 10 vertices by adding 57 edges. On average, the algorithm required 3.68 seconds of computational time for each iteration of the algorithm (i.e. for each added edge).

In the charts on the bottom, we compare the greedy algorithm with a naive algorithm that adds the edges from the k vertices with the highest closeness centrality to v. In this case we choose 10 vertices for v instead of 5. We report the ratio between the closeness value (respectively, ranking) obtained by the naive algorithm and that obtained by the greedy one. It is easy to prove that the two algorithms find the same solution for $k = 1$, while in any other case the experiments show that the greedy algorithm outperforms the naive approach. In fact, the solution computed by this latter is up to 12 times worse in terms of ranking.

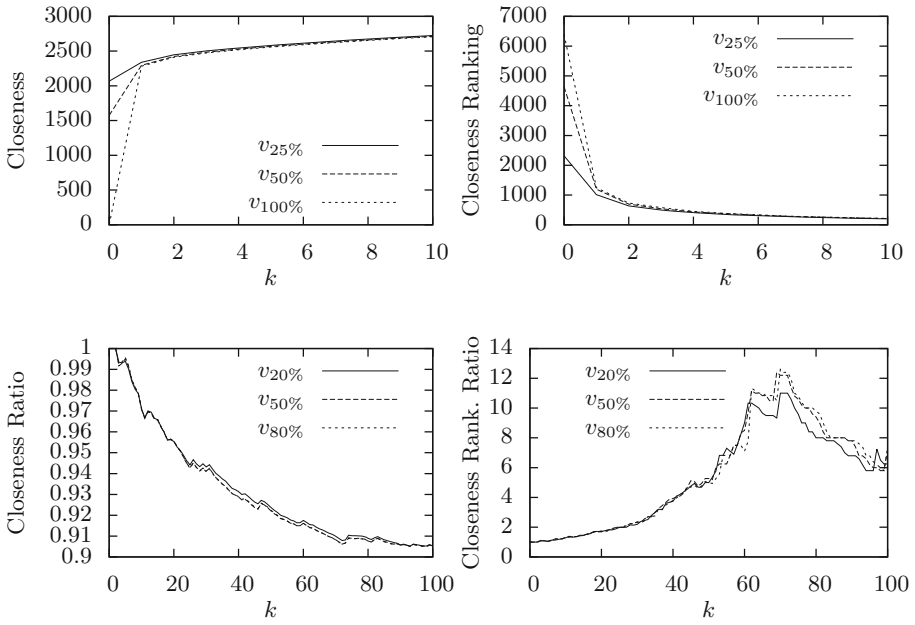

Fig. 3. Closeness centrality: (Top) performance of the greedy algorithm on network `THEORY`. (Bottom) comparison of the greedy algorithm with the naive method on network `THEORY`.

3 Improving Betweenness Centrality

Similarly to the case of the closeness centrality, we can prove, in the case of the betweenness centrality, the following two results.

Theorem 4. *Problem* MBI *cannot be approximated within a factor greater than* $1 - \frac{1}{2e}$*, unless* $P = NP$*.*

Theorem 5. *For each node* v*, function* b_v *is monotone and submodular with respect to any feasible solution for* MBI*.*

As a consequence of the previous theorem and of Theorem 1, we have that the GreedyImprovement algorithm is a $(1 - \frac{1}{e})$-approximation algorithm for the MBI problem. We now report the results of our experimental study on this algorithm. We used the same platform used for closeness and the parallel implementation of betweenness centrality of the NetworKit library [21]. First, we measured the approximation ratio of the greedy algorithm on the four types of randomly generated networks used for closeness centrality. The size of the graphs is reported in Table 2. For each combination (n, m), we generated five random graphs and used five vertices as v chosen like in the case of closeness centrality. The value of k ranges from 1 to 100.

Table 2. Betweenness centrality: comparison between the greedy algorithm and an upper bound to the optimum. The first three columns report the type and size of the graphs; the fourth (fifth, respectively) column reports the average (standard deviation, respectively) of the ratio between the value found by the greedy algorithm and the upper bound to the optimal value.

Network	$n = \|V\|$	$m = \|E\|$	**Avg.**	**Std. Dev.**
`PA`	50	65	0.9586	0.1252
`PA`	100	130	0.9500	0.1739
`ER`	50	100, 250, 500	0.7459	0.1946
`COMP`	50	100, 250, 500	0.8196	0.1946
`COPY`	50	100, 250, 500	0.8187	0.1557

In this case, we are not able to determine the optimum of MBI by means of an integer program. This is due to the non-linearity of the objective function. Therefore, in the experiments, we measured the ratio between the value of the solution found by the greedy algorithm and the optimal value of problem MBI-d, which consists in maximizing the following centrality measure:

$$d_v = \sum_{\substack{s,t \in V \\ s \neq t; s,t \neq v}} \mathbf{1}_{SP(s,t)}(v).$$

In the above formula $SP(s,t)$ denotes the set of all the vertices that belong to a shortest path from s to t and $\mathbf{1}_A(x)$ is the indicator function (i.e. $\mathbf{1}_A(x) = 1$ if $x \in A$ and $\mathbf{1}_A(x) = 0$, otherwise). It is easy to show that the optimal value of any instance of MBI-d is an upper bound for the optimal value of the corresponding MBI instance. The optimal value for MBI-d is computed by using an integer program. We solved the linear relaxation of such integer program by using GLPK. The results are reported in Table 2, where we show the average value and the standard deviation of the measured lower bound to the approximation ratio. Also in this case, the experiments show that the measured approximation ratio is by far better than the theoretical one.

For the second type of experiments we used real-world networks representing flight connection. Vertices in these networks represent airports and edges represent a connection from one airport to another. In detail, we used three networks: (i) a network obtained by crawling the EasyJet website [2] (`EasyJet` network, with 136 vertices and 1510 edges), (ii) the directed network of flights between US airports in 2010 (`USAirports` network, with 501 vertices and 5960 edges), and (iii) a network constructed from the USA Federal Aviation Administration (`USA Traffic Control` network, with 1227 vertices and 2615 edges). The last two networks are available from Konect [15]. As in the previous experiments, for each graph we used five vertices as v and we let k range from 1 to 100. The results for `EasyJet` are plotted in Fig. 4 (the results for the other networks are similar). As in the case of closeness, in the two top charts we plot the betweenness centrality and the ranking of node v as a function of k, in the two bottom charts,

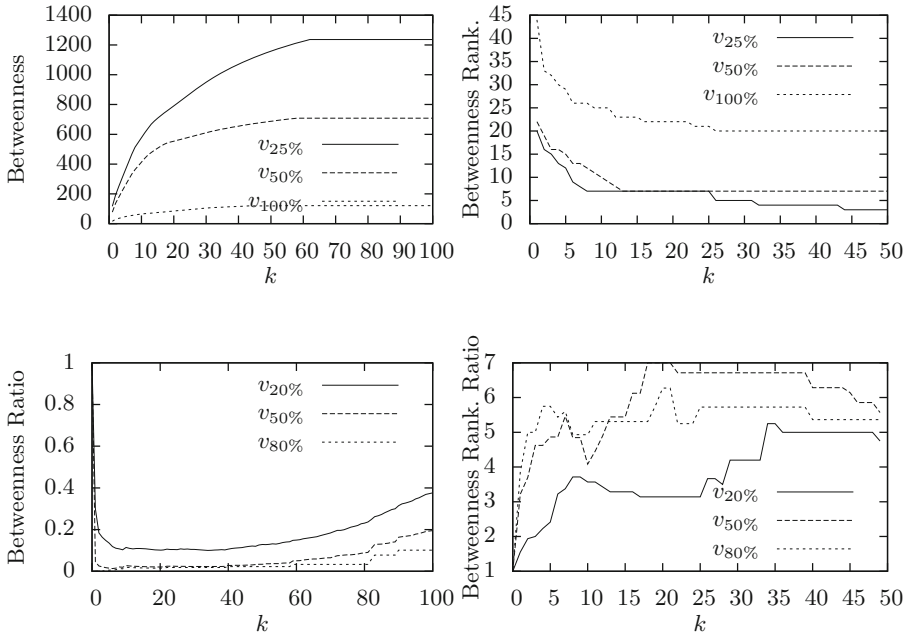

Fig. 4. Betweenness centrality: (Top) Performance of the greedy algorithm on network `EasyJet`; (Bottom) Comparison of the greedy algorithm with the naive method on network `EasyJet`

we compare the greedy algorithm with the naive algorithm. Similar results as for closeness can be observed. However, in this case, the improvement in value and in ranking is smaller than in the case of closeness. This is due to the fact that we only add incoming edges while the number of shortest paths passing through v also depends on the edges outgoing from v. We leave the problem of adding both incoming and outgoing edges as an open problem. Also in this case our algorithm outperforms the naive approach by computing solutions that are up to 7 times better in terms of ranking. On average, the algorithm required 0.33 seconds of computational time for each iteration of the algorithm (i.e. for each added edge).

4 Conclusion and Future Research

In this paper, we have proposed a greedy approximation algorithm for efficiently computing a set of edges that a node can decide to add to a graph in order to increase its betweenness or closeness centrality. The algorithm has been tested on several relatively small random graphs and, then, applied to several real-world collaboration networks. As future works, we plan to extend our approach to weighted graphs and to other centrality measures, to analyze a generalization of

the problem considered in this paper (by allowing the addition of edges incident to other vertices), to study the problem of maximizing the ranking improvement (instead of the centrality value), to apply algorithmic game theoretical techniques (in order to deal with the concurrent addition of edges by different vertices of the graph), and dynamic algorithm techniques (in order to make the greedy algorithm more efficient).

References

1. Arnetminer (accessed: 2015–01-15). http://arnetminer.org
2. Easyjet (accessed: 2015–01-15). http://www.easyjet.com
3. GLPK - GNU Linear Programming Kit. http://www.gnu.org/software/glpk
4. Avrachenkov, K., Litvak, N.: The effect of new links on google pagerank. Stoc. Models **22**(2), 319–331 (2006)
5. Boldi, P., Vigna, S.: Axioms for centrality. Internet Math. **10**(3–4), 222–262 (2014)
6. Bollobás, B., Borgs, C., Chayes, J., Riordan, O.: Directed scale-free graphs. In: Proc. of the 14th Annu. ACM-SIAM Symp. on Disc. Alg. (SODA), pp. 132–139. SIAM (2003)
7. Brandes, U.: A faster algorithm for betweenness centrality. J. Math. Sociol. **25**(2), 163–177 (2001)
8. Chierichetti, F., Kumar, R., Lattanzi, S., Panconesi, A., Raghavan, P.: Models for the compressible web. In: Proc. of the 50th Annu. Symp. on Found. of Comput. Sci. (FOCS), pp. 331–340. IEEE (2009)
9. Cohen, E., Delling, D., Pajor, T., Werneck, R.F.: Computing classic closeness centrality, at scale. Technical Report MSR-TR-2014-71 (2014)
10. Demaine, E.D., Zadimoghaddam, M.: Minimizing the diameter of a network using shortcut edges. In: Kaplan, H. (ed.) SWAT 2010. LNCS, vol. 6139, pp. 420–431. Springer, Heidelberg (2010)
11. Erdős, P., Rényi, A.: On random graphs I. Publ. Math. **6**, 290–297 (1959)
12. Feige, U.: A threshold of ln n for approximating set cover. J. ACM **45**(4) (1998)
13. Ishakian, V., Erdös, D., Terzi, E., Bestavros, A.: A framework for the evaluation and management of network centrality. In: Proc. of the 12th SIAM Int. Conf. on Data Mining (SDM), pp. 427–438. SIAM (2012)
14. Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., Upfal, E.: Stochastic models for the web graph. In: Proc. of the 41st Annu. Symp. on Found. of Comput. Sci. (FOCS), pp. 57–65. IEEE (2000)
15. Kunegis, J.: KONECT - The Koblenz network collection. In: Proc. of the 1st Int. Web Observatory Work. (WOW), pp. 1343–1350 (2013)
16. Malighetti, P., Martini, G., Paleari, S., Redondi, R.: The impacts of airport centrality in the EU network and inter-airport competition on airport efficiency. Technical Report MPRA-7673 (2009)
17. Meyerson, A., Tagiku, B.: Minimizing average shortest path distances via shortcut edge addition. In: Dinur, I., Jansen, K., Naor, J., Rolim, J. (eds.) Approximation, Randomization, and Combinatorial Optimization. LNCS, vol. 5687, pp. 272–285. Springer, Heidelberg (2009)
18. Nemhauser, G., Wolsey, L., Fisher, M.: An analysis of approximations for maximizing submodular set functions-I. Math. Program. **14**(1), 265–294 (1978)
19. Olsen, M., Viglas, A.: On the approximability of the link building problem. Theor. Comput. Sci. **518**, 96–116 (2014)

20. Riondato, M., Kornaropoulos, E.M.: Fast approximation of betweenness centrality through sampling. In: Proc. of the 7th ACM Int. Conf. on Web Search and Data Mining, (WSDM), pp. 413–422. ACM (2014)
21. Staudt, C.L., Sazonovs, A., Meyerhenke, H.: Networkit: An interactive tool suite for high-performance network analysis. arXiv preprint arXiv:1403.3005 (2014)
22. Williamson, D., Shmoys, D.: The Design of Approximation Algorithms. Cambridge University Press (2011)
23. Yan, E., Ding, Y.: Applying centrality measures to impact analysis: A coauthorship network analysis. J. Ass. Inf. Sci. Tech. **60**(10), 2107–2118 (2009)